COL380 Assignment 2: MPI

By Mohit Sharma(2019CS10372)

Idea:

1) Each rank reads data from file in parallel and create its adjacency list.

2) Then we sort nodes based on (number of neighbours, node id).

This helps give nodes with equally large numbers of neighbours to each graph, otherwise one rank could work with nodes with all high number of neighbours and other with 0 number of neighbours, so second node will waste time in that case.

3) Each rank writes its calculated recommendation in the output file at correct offset parallelly with other ranks.

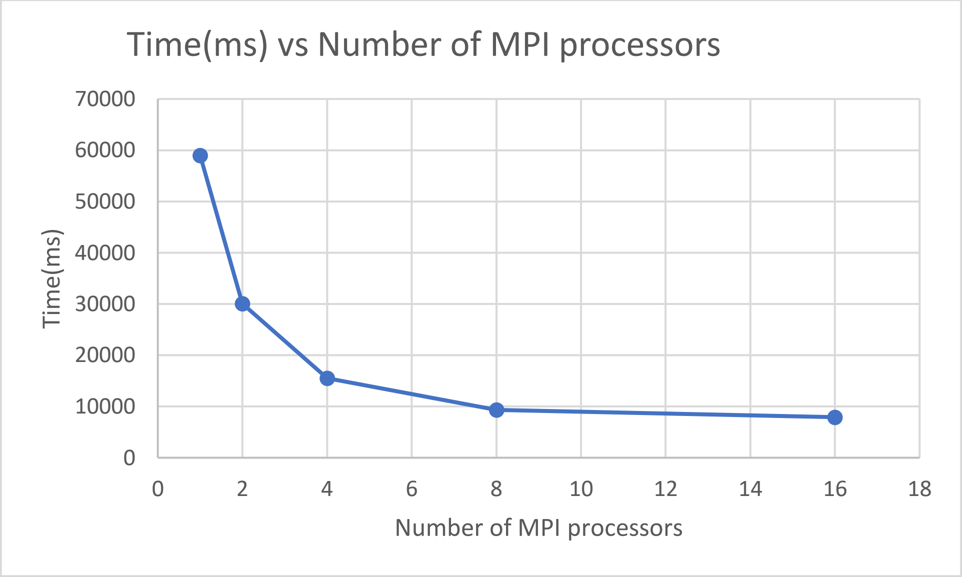
Common Parameters used for analysis:

Restart probability: 0.1	Number of steps: 5000
Number of walks: 30	Number of recommendations: 20
Seed: 369	

1) Data 1

Number of nodes = 8717

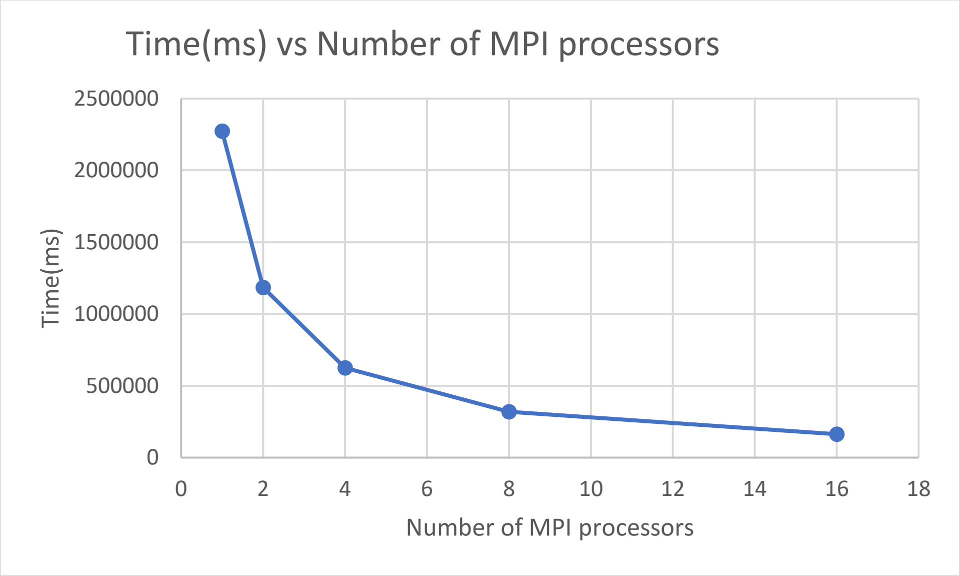
Number of edges = 31525



2) Data 2

Number of nodes = 81867

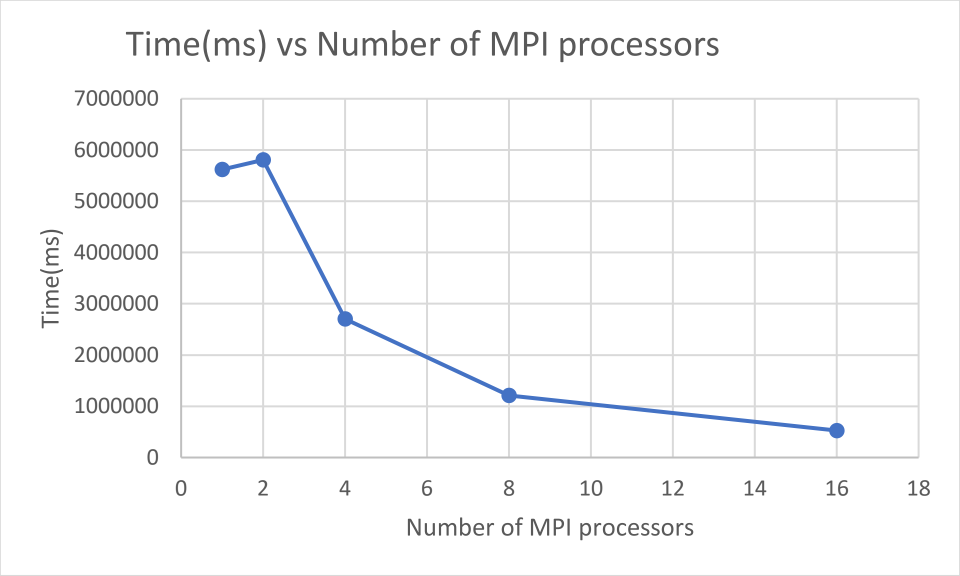
Number of edges = 545671



3) Data 3

Number of nodes = 82168

Number of edges = 870161



Old Idea: (Not calculated influence score of each node in a separate rank) (Not good idea)

1) Each rank reads data from file in parallel and create its adjacency list.

2) For every node u, we have to run the random walk (number of children) * (number of walks) times.

Where, number of children is the outgoing edges in the graph

So, I divided these total number of walks to be done equally among the ranks.

Each rank then calculates influence score of all the nodes for given node u.

Then I used MPI reduce to sum the influence score across all nodes which is then processed by rank 0 to get top number of recommendations and then it is written in the output.dat file.

This idea wastes time when all processors send there data to rank 0 and then it does the recommendation calculation and printing and other just sit and wait for it to gain receive new data.

Common Parameters used for analysis:

Restart probability: 0.1 Number of steps: 5000

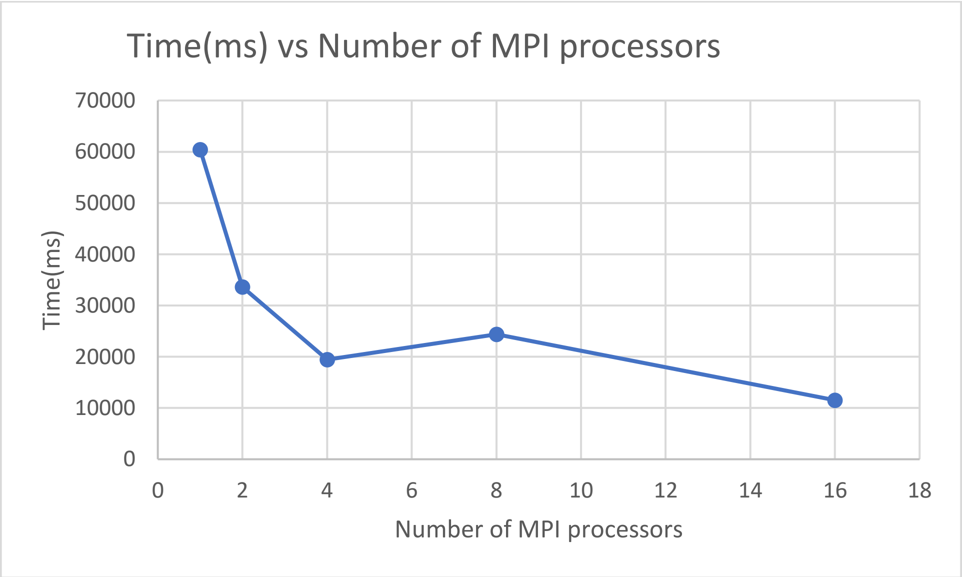
Number of walks: 30 Number of recommendations: 20

Seed: 369

1) Data 1

Number of nodes = 8717

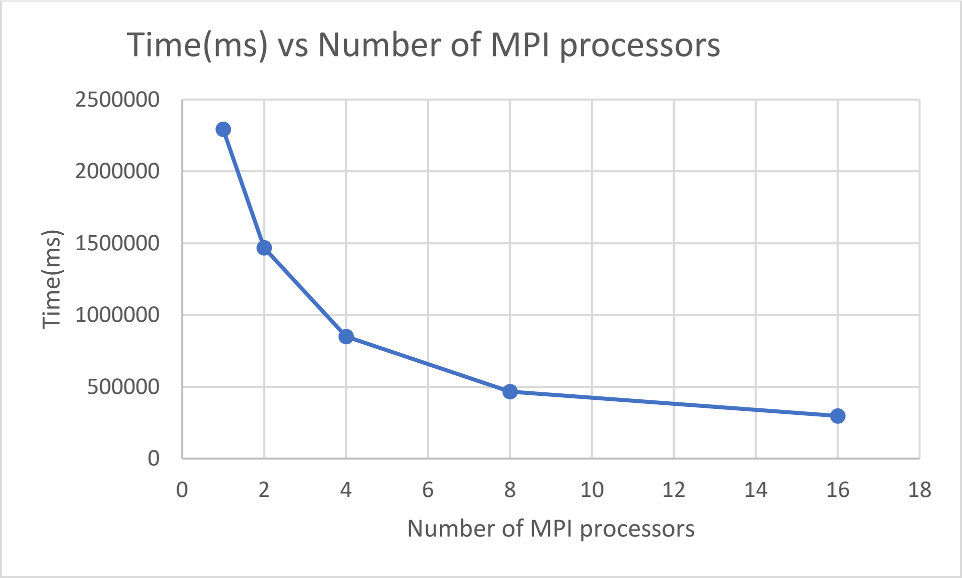
Number of edges = 31525



2) Data 2

Number of nodes = 81867

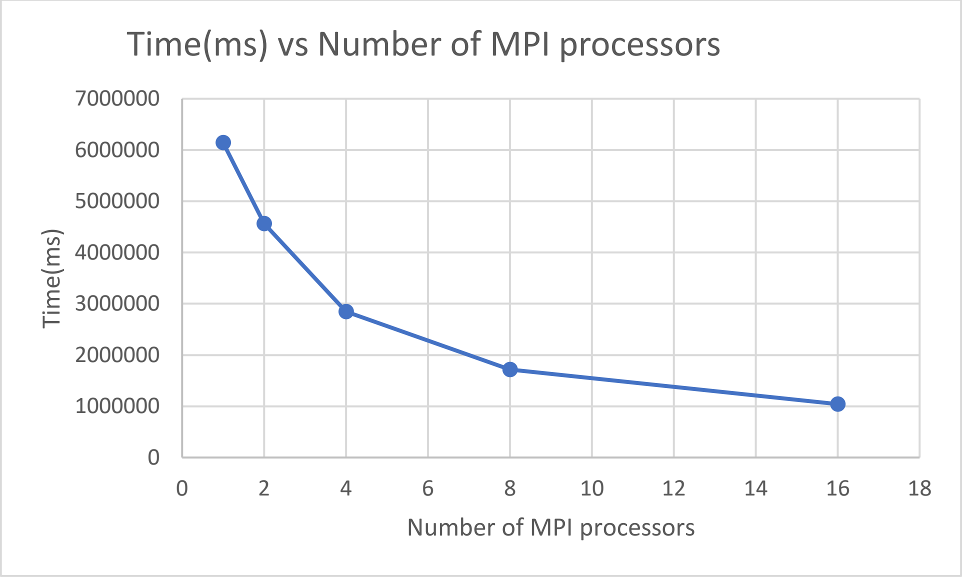
Number of edges = 545671



3) Data 3

Number of nodes = 82168

Number of edges = 870161


Time(ms) vs Number of MPI processors
Time(ms)
Number of MPI processors